Streaming Astronomical Signal Processing on Next Generation HPC Hardware

Chris Broekema

ASTRON
(Netherlands Foundation for Research in Astronomy)

Streaming real-time Astronomical Signal Processing on HPC Hardware

Chris Broekema

ASTRON
(Netherlands Foundation for Research in Astronomy)

ASTRON

Making astronomical discoveries happen through innovative observing facilities

the LOFAR idea

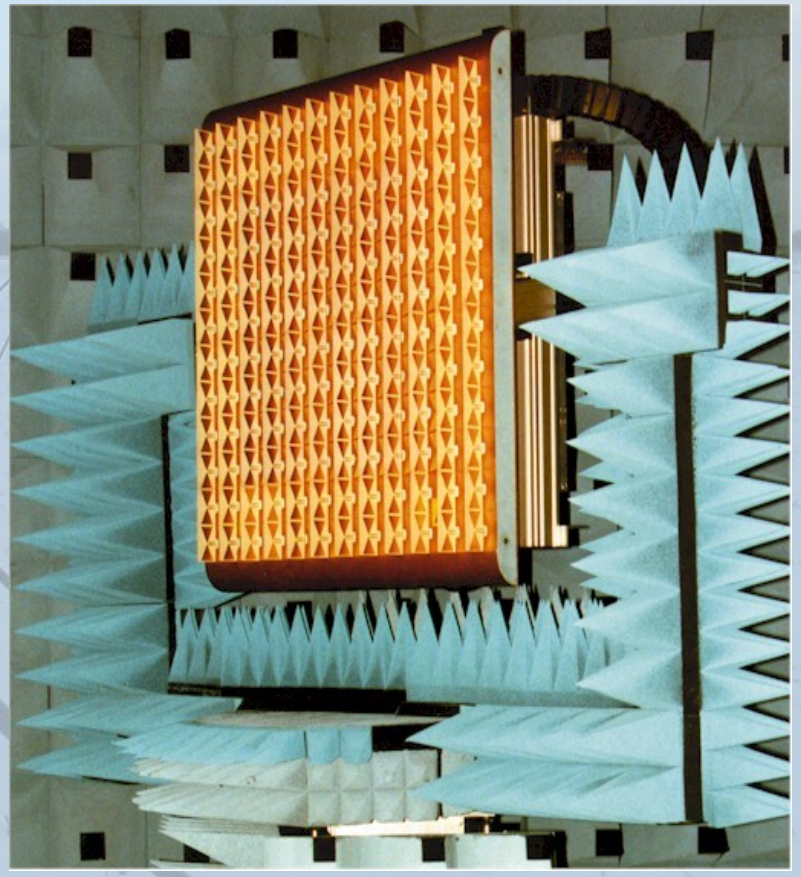
- Science community: desire to look further into the past (higher red-shifts)
- ASTRON: realization that Moore's law would enable us to build a cost effective low frequency array (eventually)
- Furthermore, the LOFAR frequency range (10 – 200 Mhz) is mostly unexplored: huge potential for scientific discovery
- Use phased array technology to create relatively cheap, large telescopes which can be combined into one huge synthesized radio telescope

LOFAR Technical concept

- LOFAR is a very large distributed radiotelescope:
  - ~15.000 small antennas (130.000 dipoles per polarization)
  - in 77 stations in NL, 10 – 15 foreign
  - 40 Tbit/sec raw data
  - 40 Tflop central processing
  - innovative software systems
  - datamining and visualisation
- Full and exclusive control via the internet
- Instantaneous view of the entire sky, several simultaneous users
- Extensions to 1000km baseline

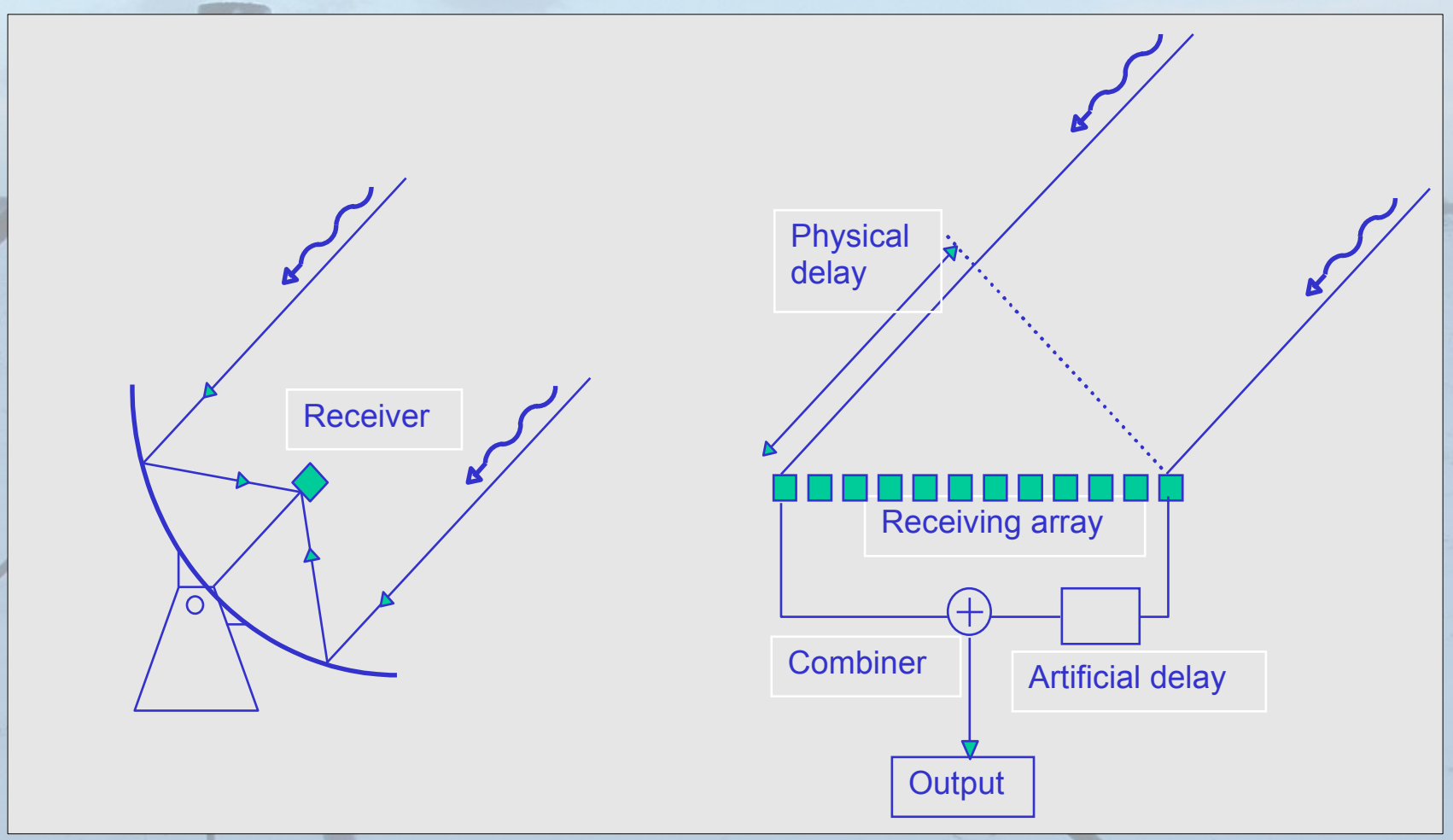
Shift from steel to silicon

Parabolic dish to Phased array



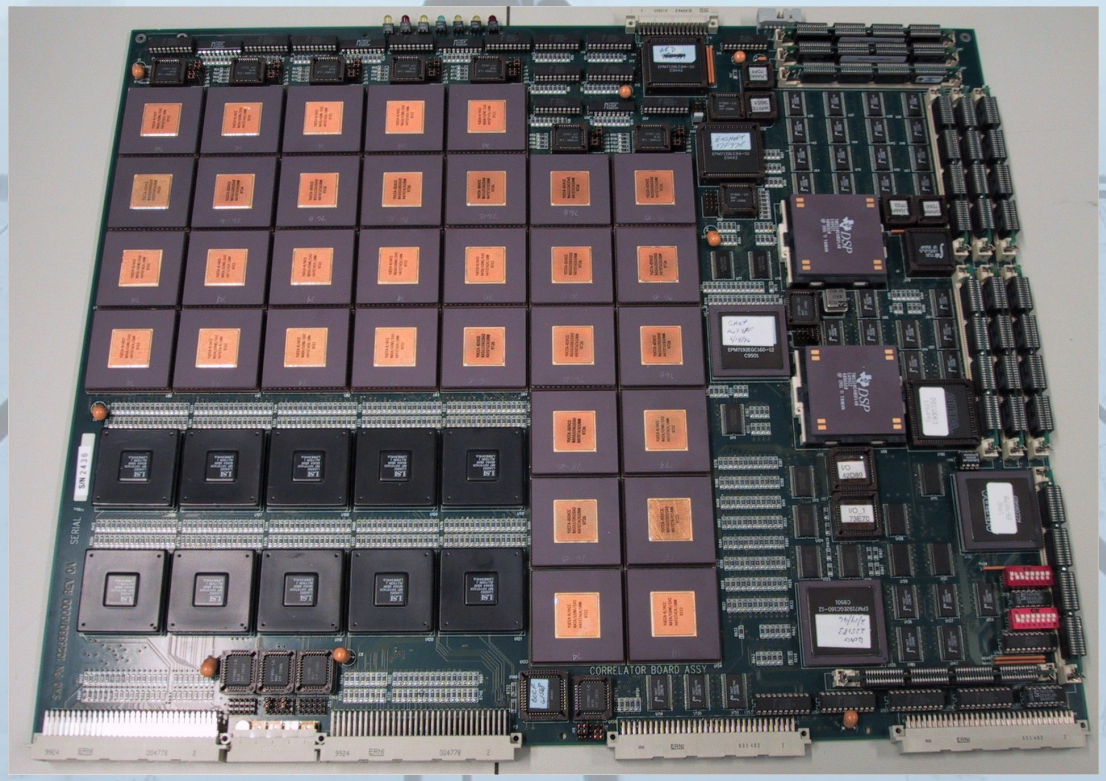
Effelsberg(D) radiotelescope, d = 100m

Reflectors vs. Phased arrays



Shift from steel to silicon

Custom VLSI hardware to COTS HPC



LOFAR

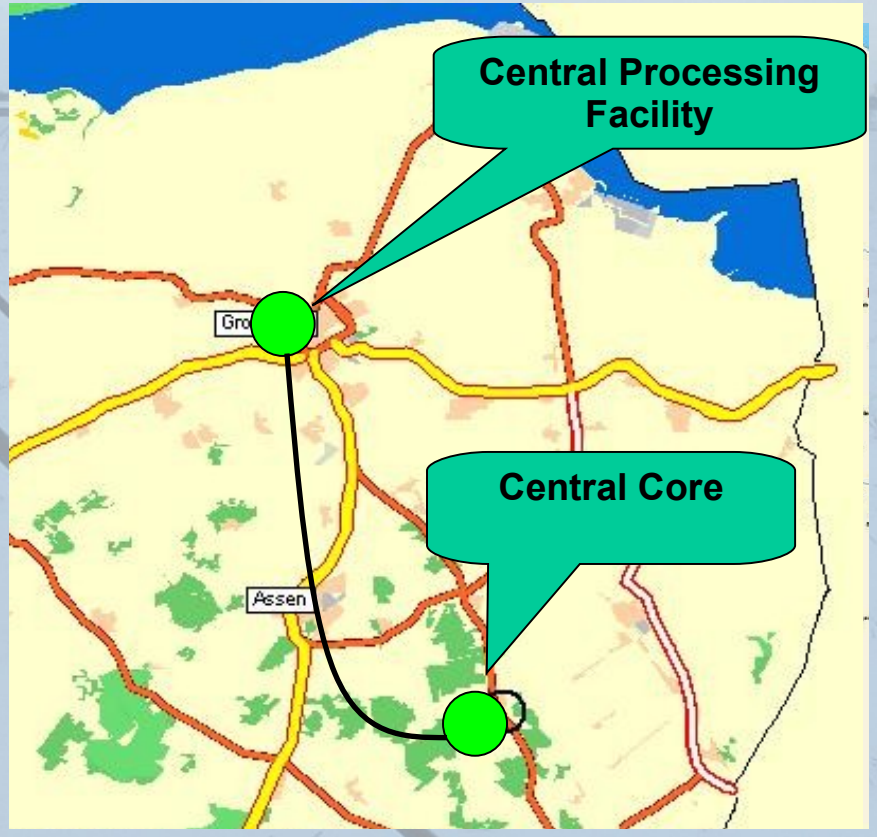
ASTRON

LOFAR sensors

- Low band antenna: 20–80 MHz
- High band antenna: 110-240 MHz
- Geophones
- high precision agriculture
- Infrasound detectors

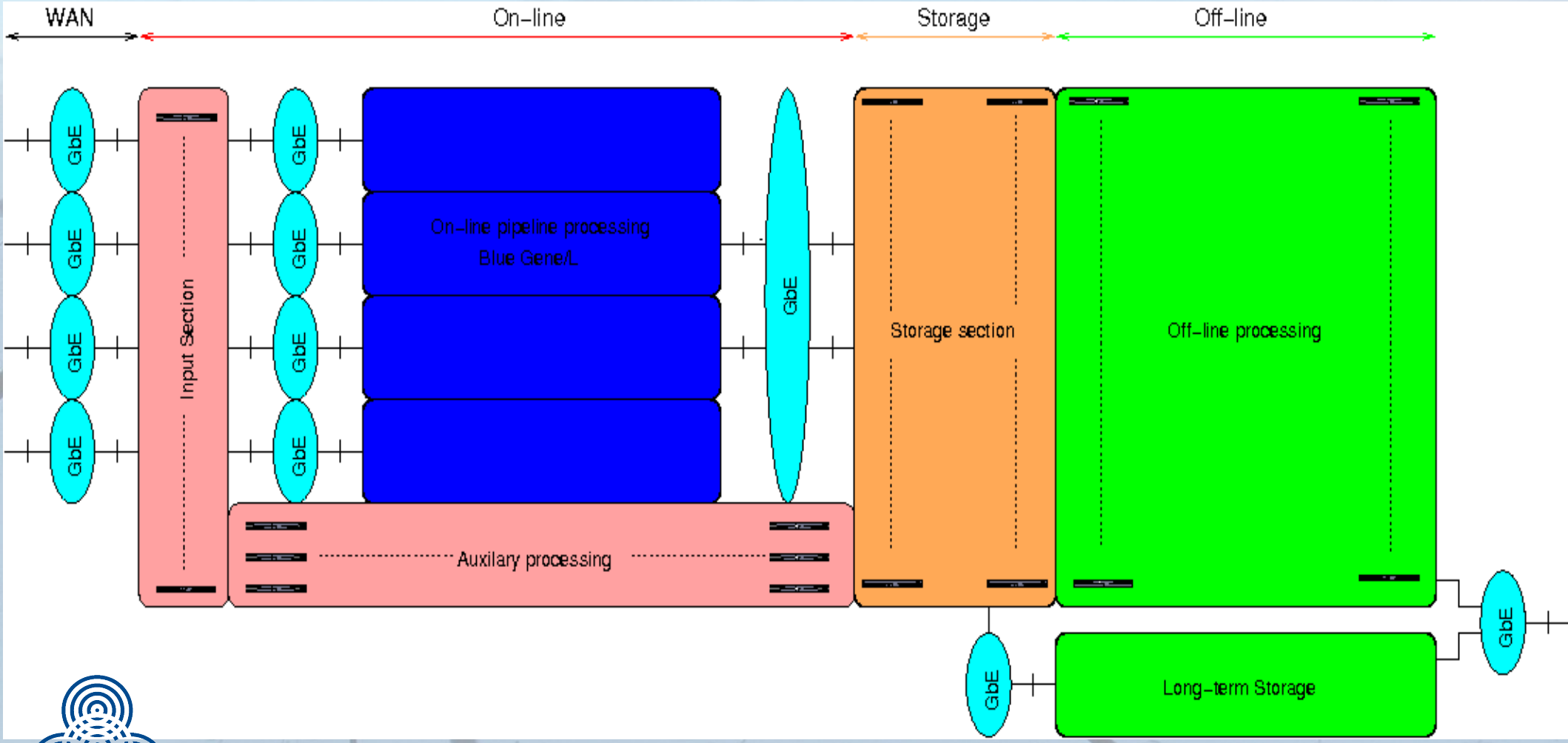
Wide Area Network

- Data transport from stations and central core to central processor facility
- Dedicated fiber connection between core and central processor
  - up to 800 Gbps bandwidth
  - 10 GbE CWDM
  - 8 channels
  - length ~70 km
- Remote station connections:
  - 1 GbE technology (up to 10 GbE)
  - data rate 2 Gbps from antennas + monitoring + other sensors



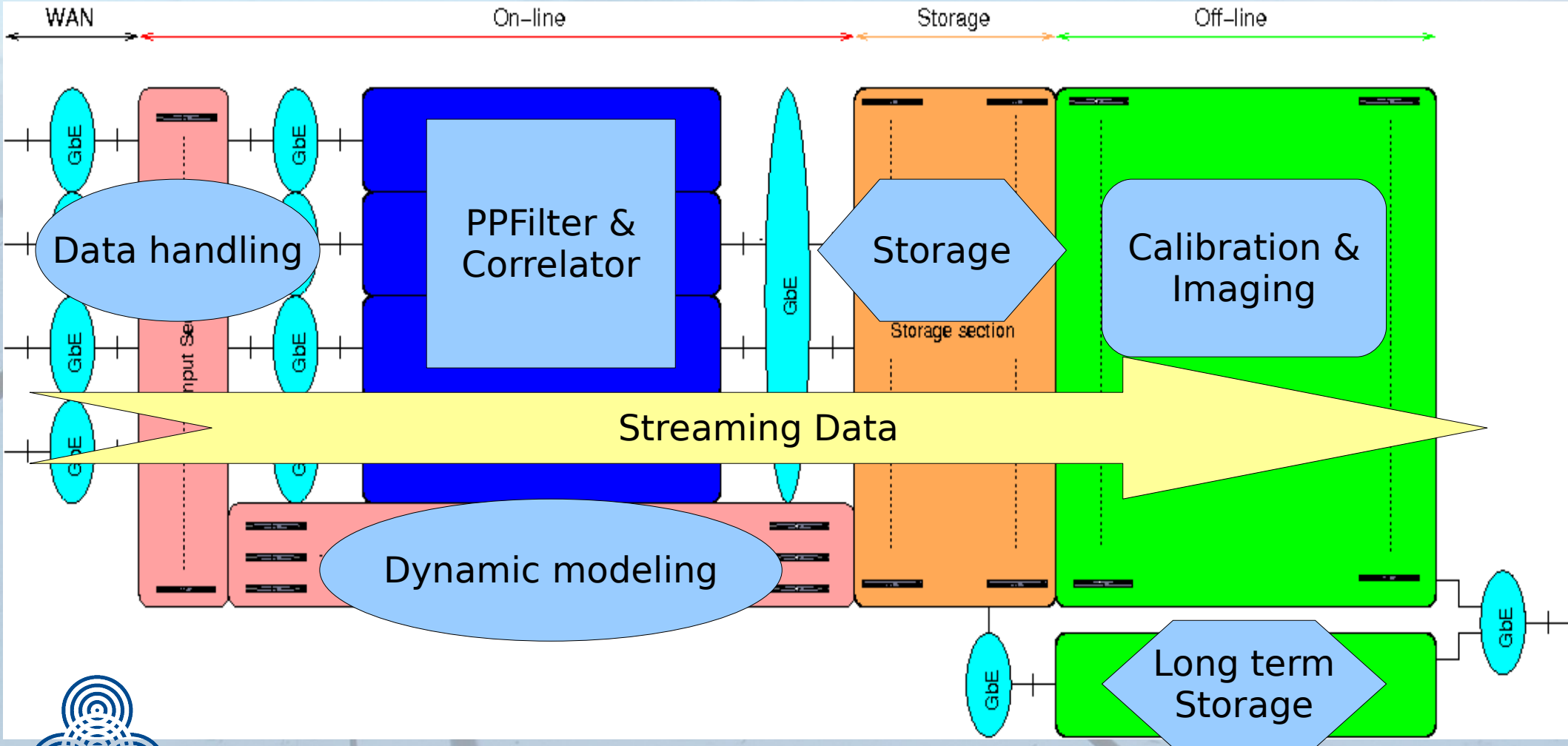
CEP implementation model

Multiple Linux based sub-clusters, Blue Gene/L inside

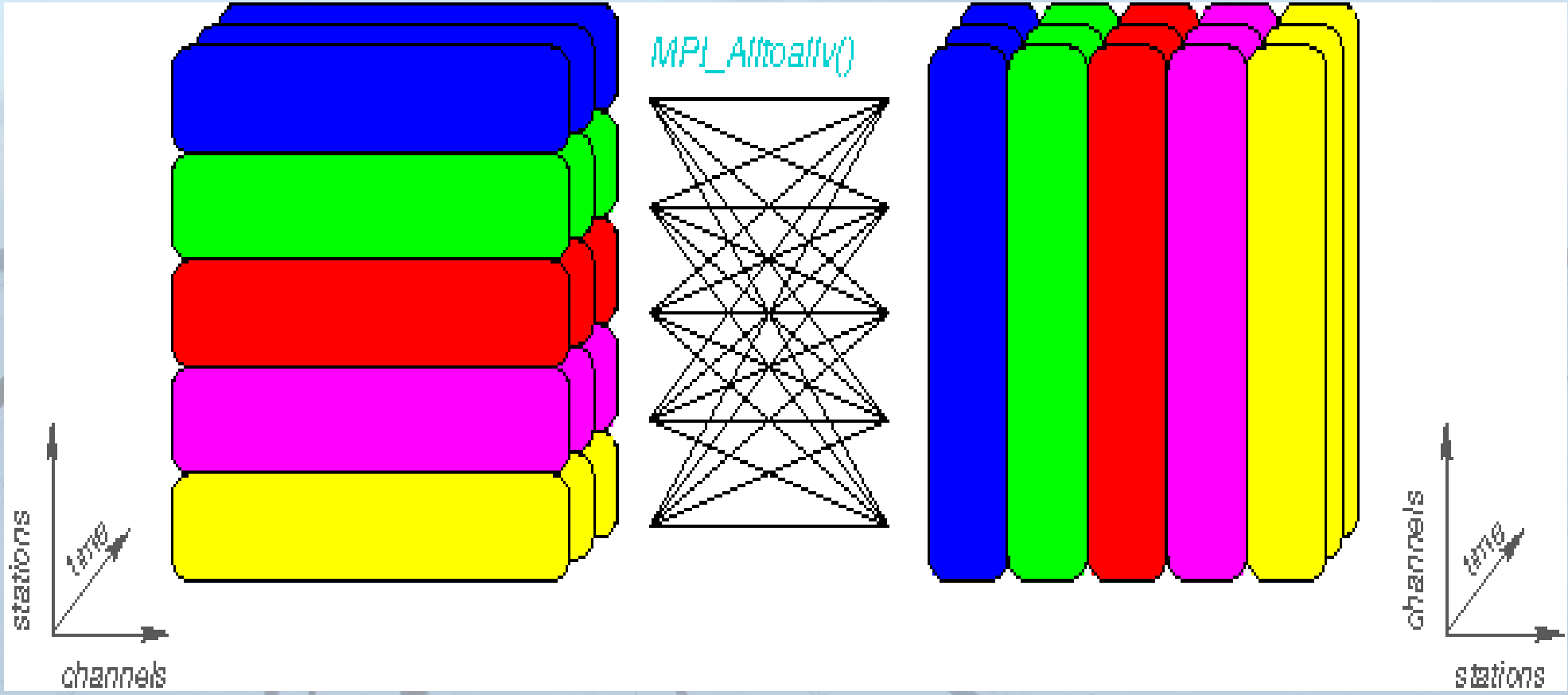


CEP implementation model

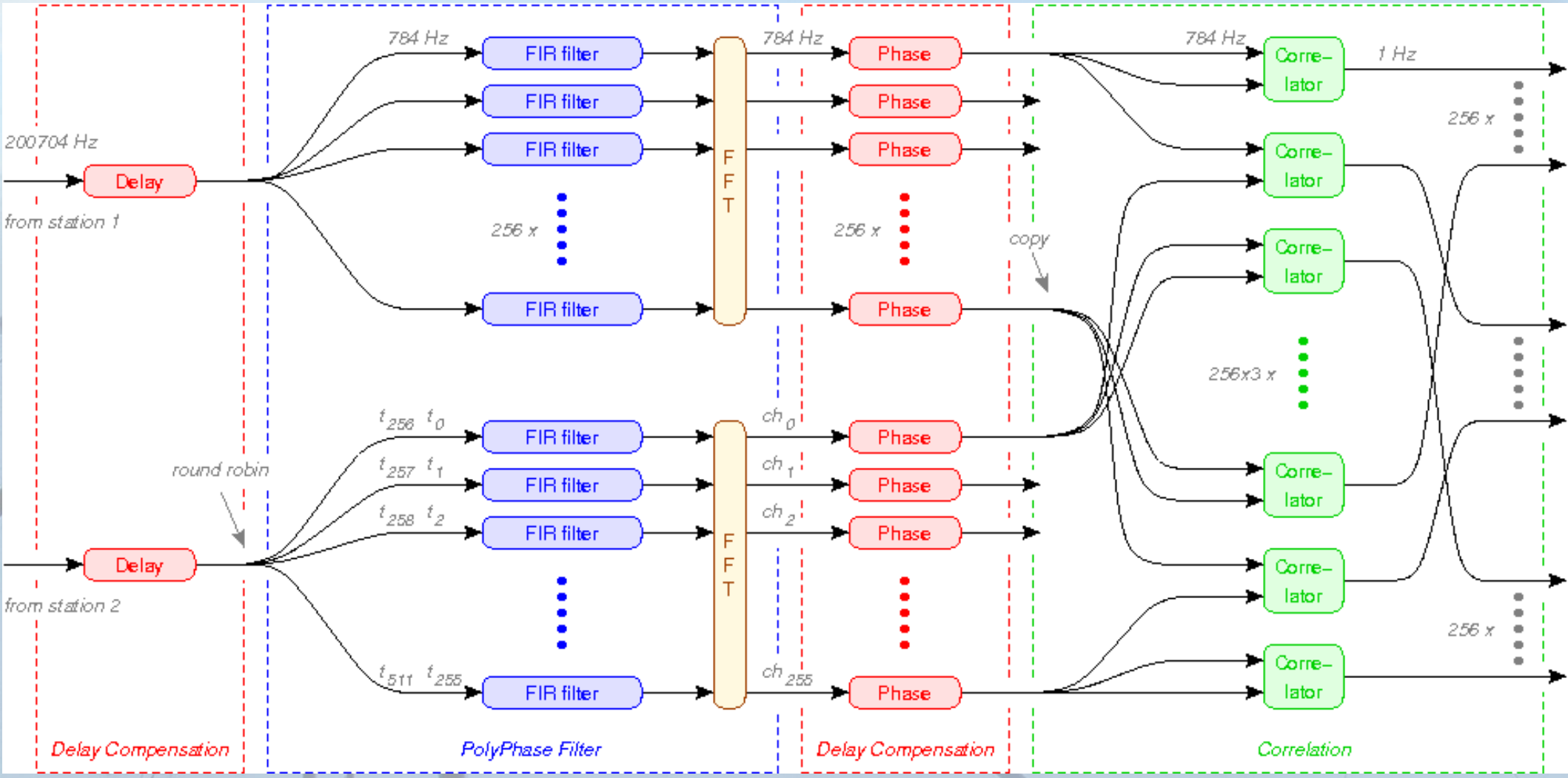
Multiple Linux based sub-clusters, Blue Gene/L inside



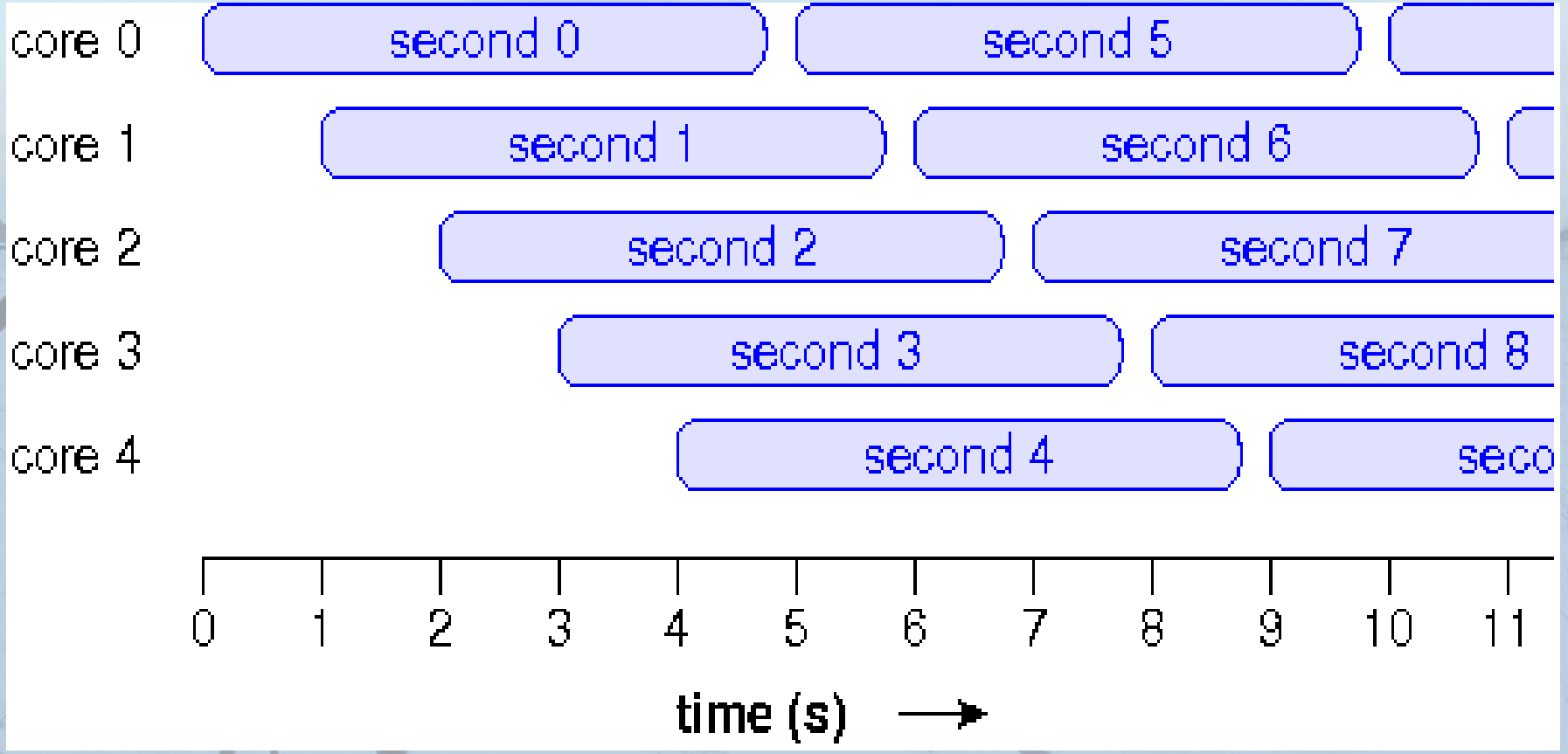
LOFAR Network transpose



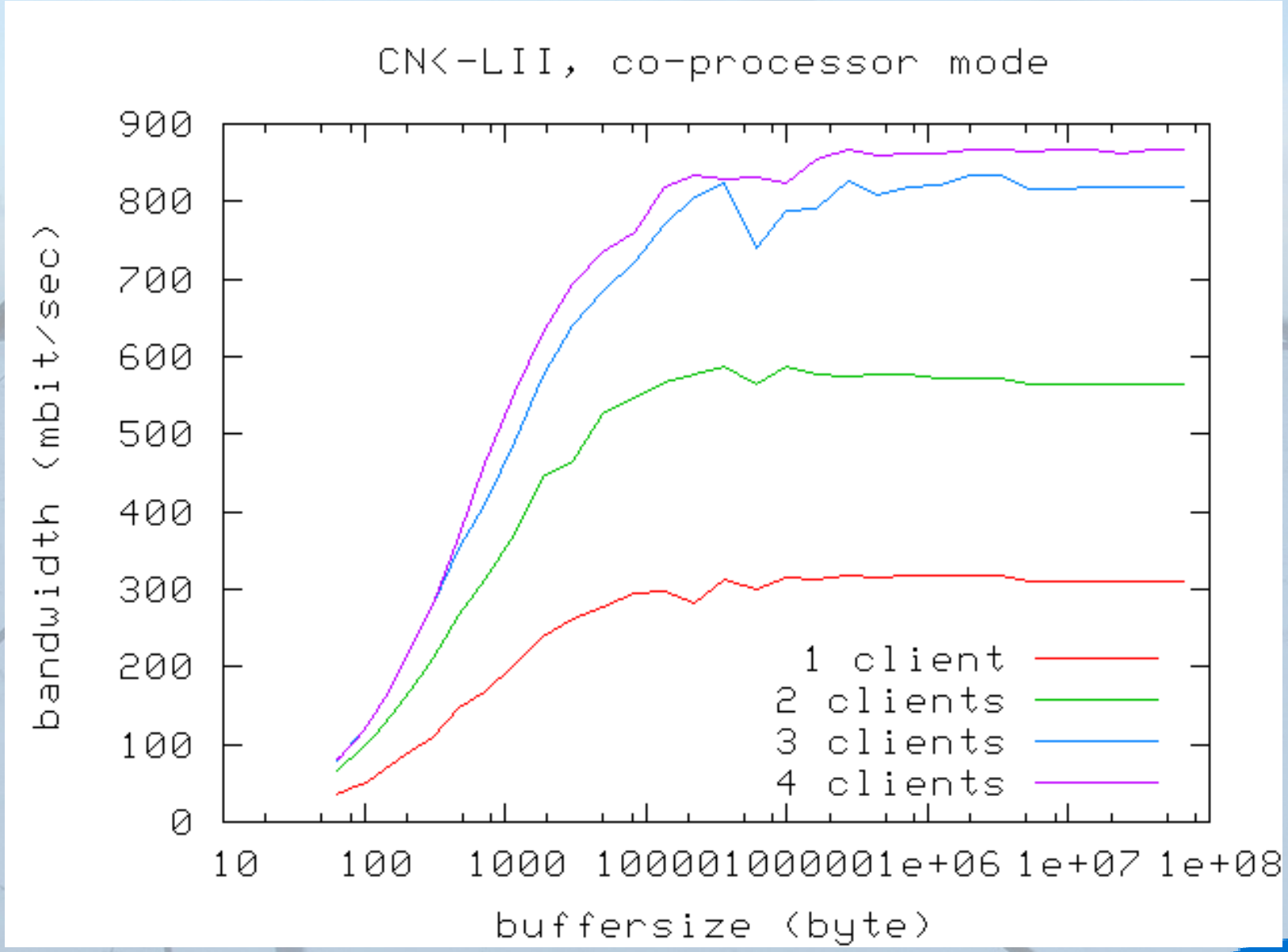
LOFAR CEP Processing



LOFAR CEP Processing



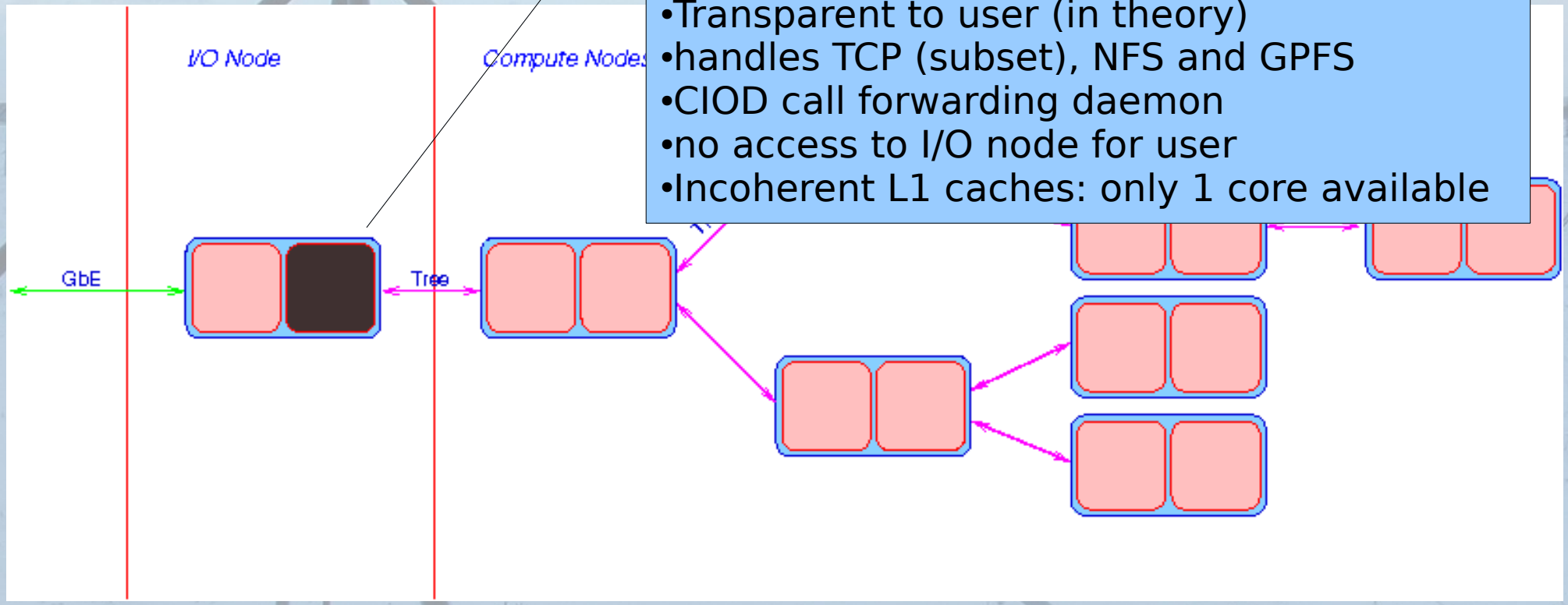
Blue Gene/L I/O



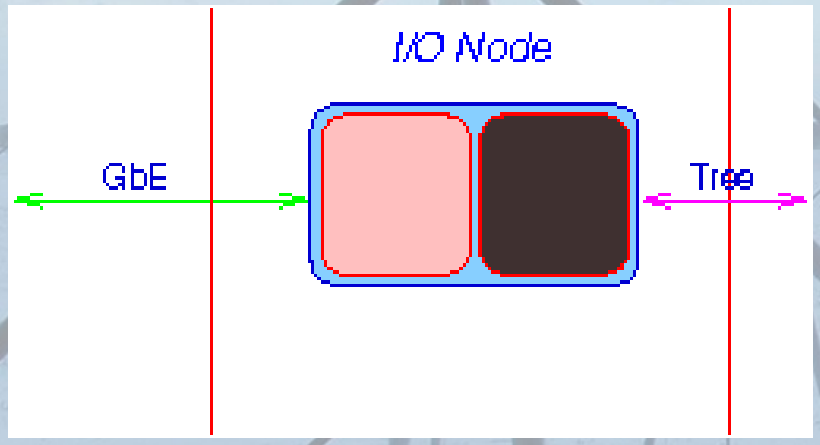
Blue Gene/L I/O

Default IBM software:

- Linux 2.6 kernel (originally 2.4)
- Transparent to user (in theory)
- handles TCP (subset), NFS and GPFS
- CIOD call forwarding daemon
- no access to I/O node for user
- Incoherent L1 caches: only 1 core available



Blue Gene/L I/O



Enter ZeptoOS

Drop-in replacement for IBM kernel image
Project by Argonne National Lab (ANL)
Provides:
- sshd, so users can log into I/O node
- profiling & debugging tools
- GNU based toolchain for application building
- PVFS2 support

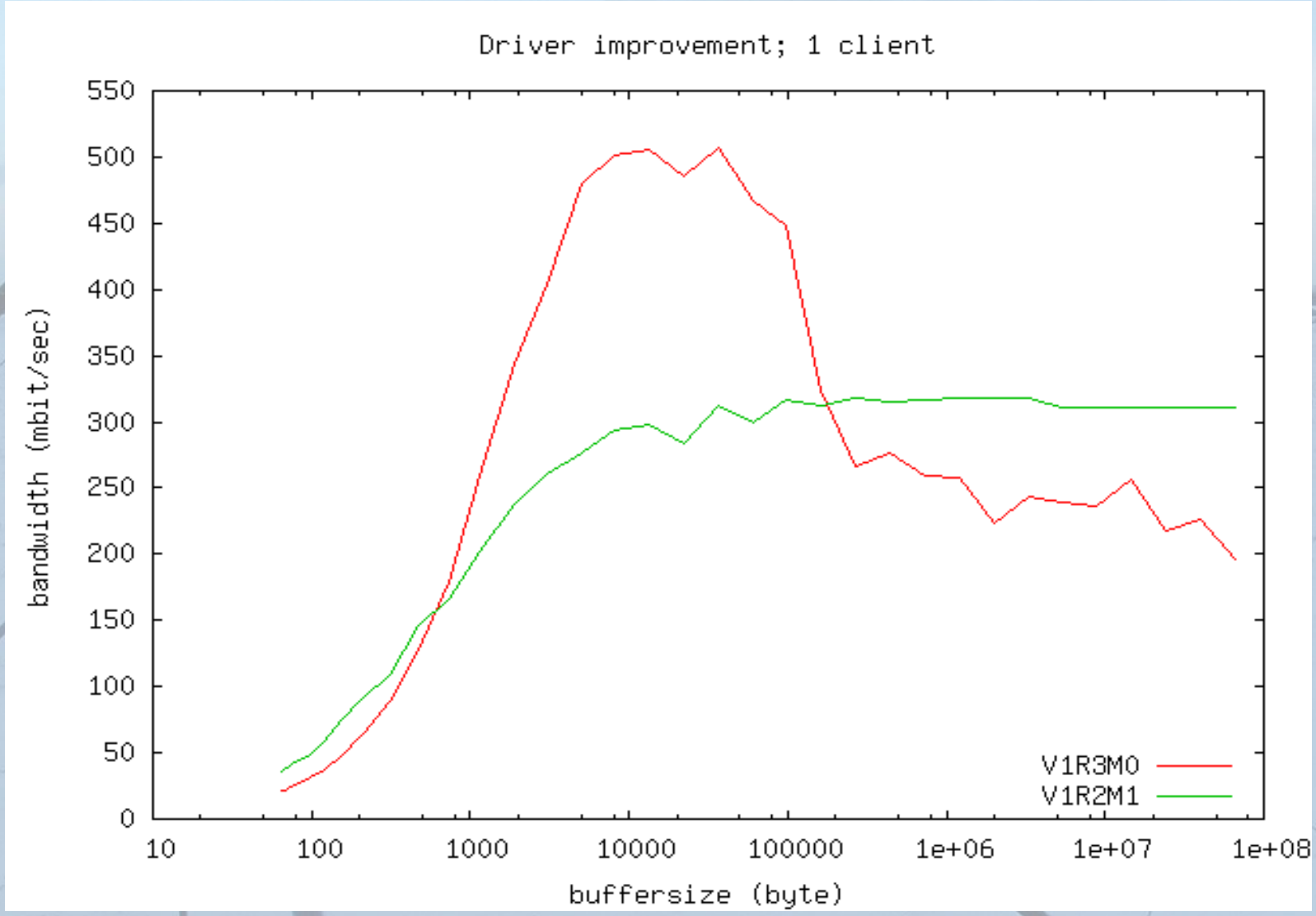
ZeptoOS gave us the ability to run applications on the I/O node.

We've just increased our Blue Gene/L by by 768 nodes, but they're only single core not part of the Blue Gene/L internal infrastructure.

Blue Gene/L I/O

- ZeptoOS allowed us to experiment on ION
- Problem exploration with IBM Rochester
- A fix was introduced in driver V1R3M0
- Performance improvement 1 connection
- Slight decrease multiple connections
- But, still not enough throughput for LOFAR!

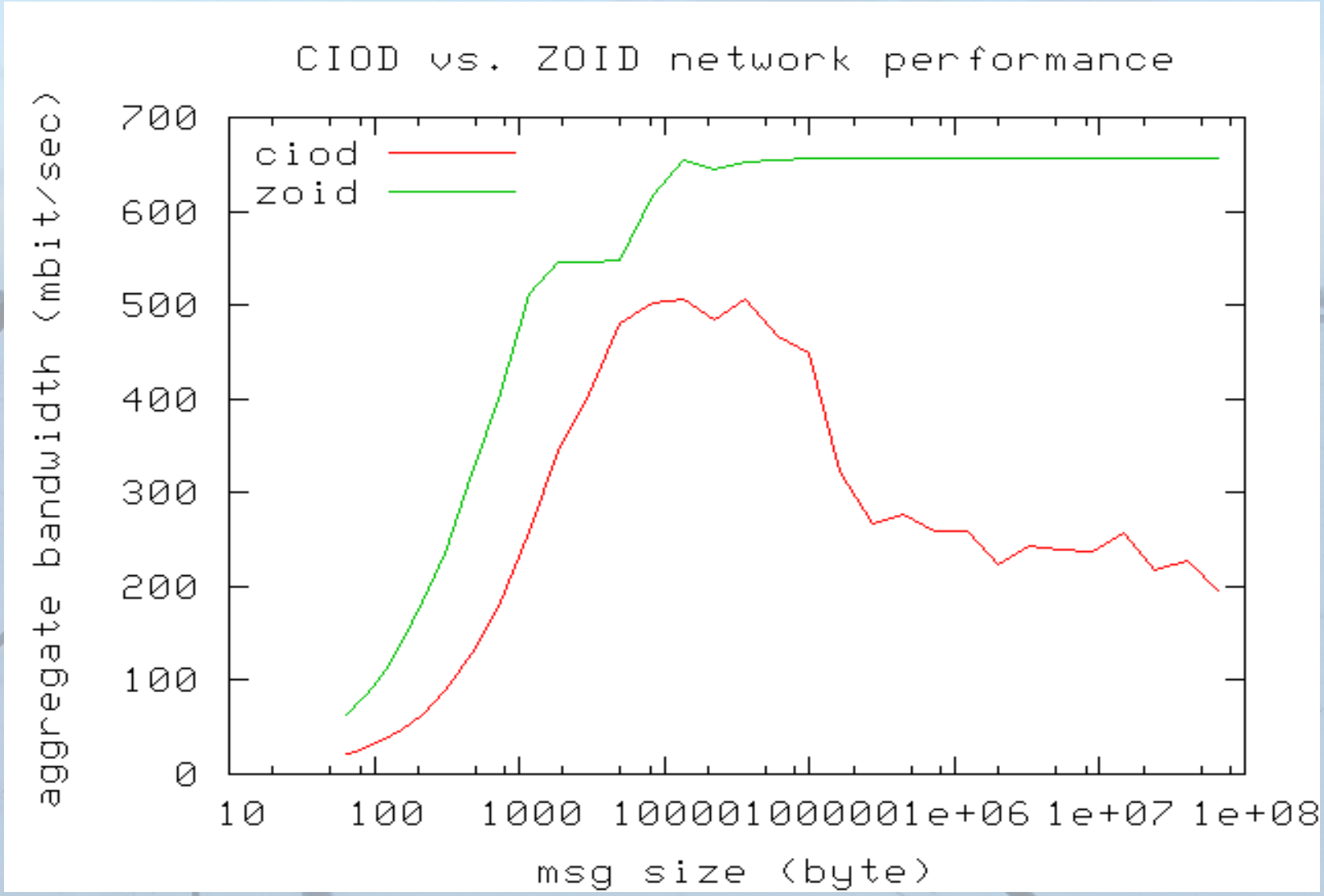
Blue Gene/L I/O



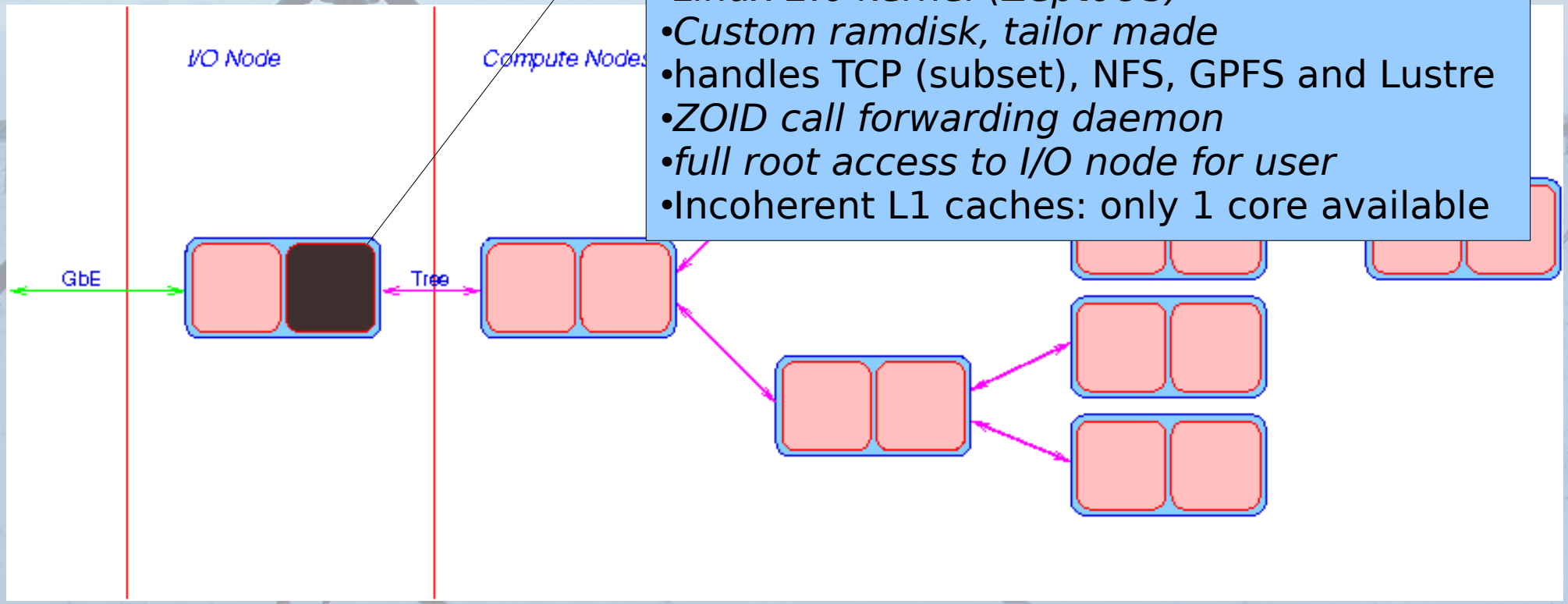
Blue Gene/L I/O

- ZOID function call forwarding daemon
- Drop-in replacement for CIOD
- Goal: improve upon CIOD performance
- Research project started by ANL, now a joint project ANL / ASTRON
- Source Code available
- Easily extensible
- 33 – 50% more subbands using ZOID

Blue Gene/L I/O



Blue Gene/L I/O



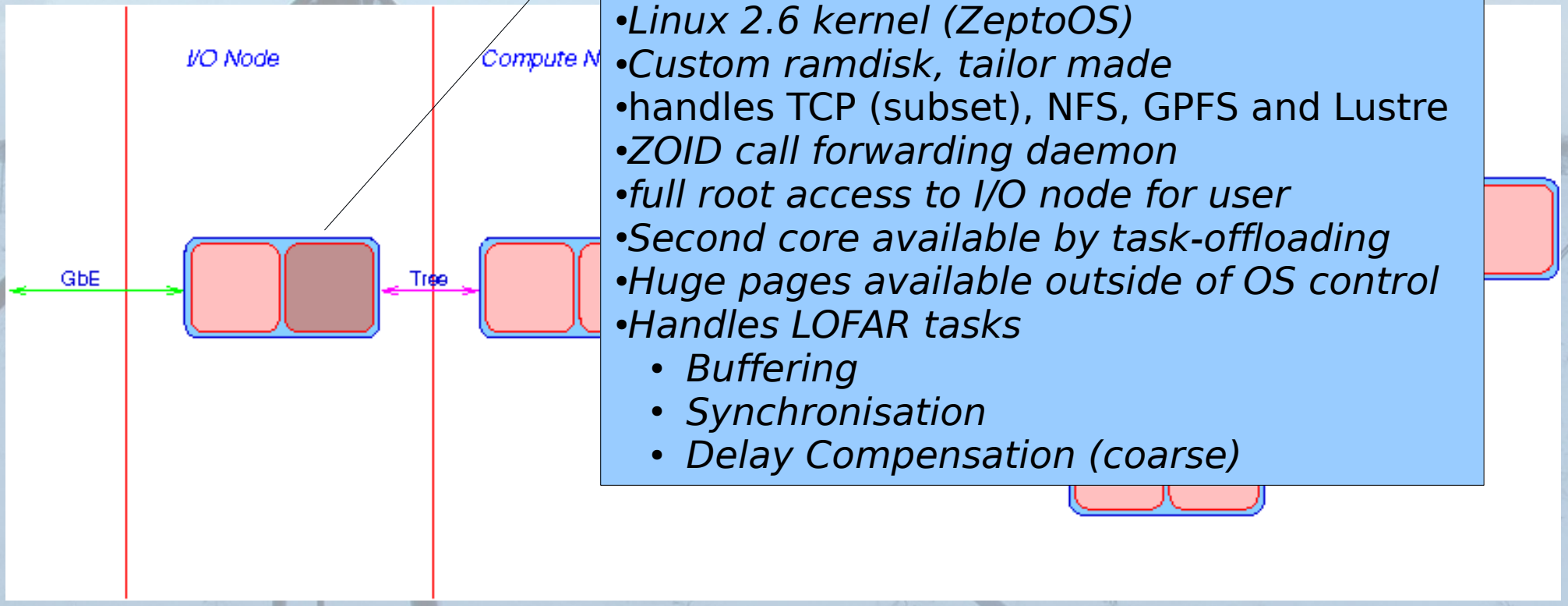
ZeptoOS & ZOID:

- *Linux 2.6 kernel (ZeptoOS)*
- *Custom ramdisk, tailor made*
- handles TCP (subset), NFS, GPFS and Lustre
- *ZOID call forwarding daemon*
- *full root access to I/O node for user*
- Incoherent L1 caches: only 1 core available

Blue Gene/L I/O

- We now have full control over I/O node
- Can we do extra processing on the I/O node without sacrificing performance?
  - The I/O node is underpowered as it is
  - Only one core available
  - CPU not optimised for Linux
  - Limited memory buffer Ethernet device
  - But, the available computing power is tempting
  - Might replace or reduce inputsection requirement

Blue Gene/L I/O



ZeptoOS & LOFAR specific ZOID:

- *Linux 2.6 kernel (ZeptoOS)*
- *Custom ramdisk, tailor made*
- handles TCP (subset), NFS, GPFS and Lustre
- *ZOID call forwarding daemon*
- *full root access to I/O node for user*
- *Second core available by task-offloading*
- *Huge pages available outside of OS control*
- *Handles LOFAR tasks*
  - *Buffering*
  - *Synchronisation*
  - *Delay Compensation (coarse)*

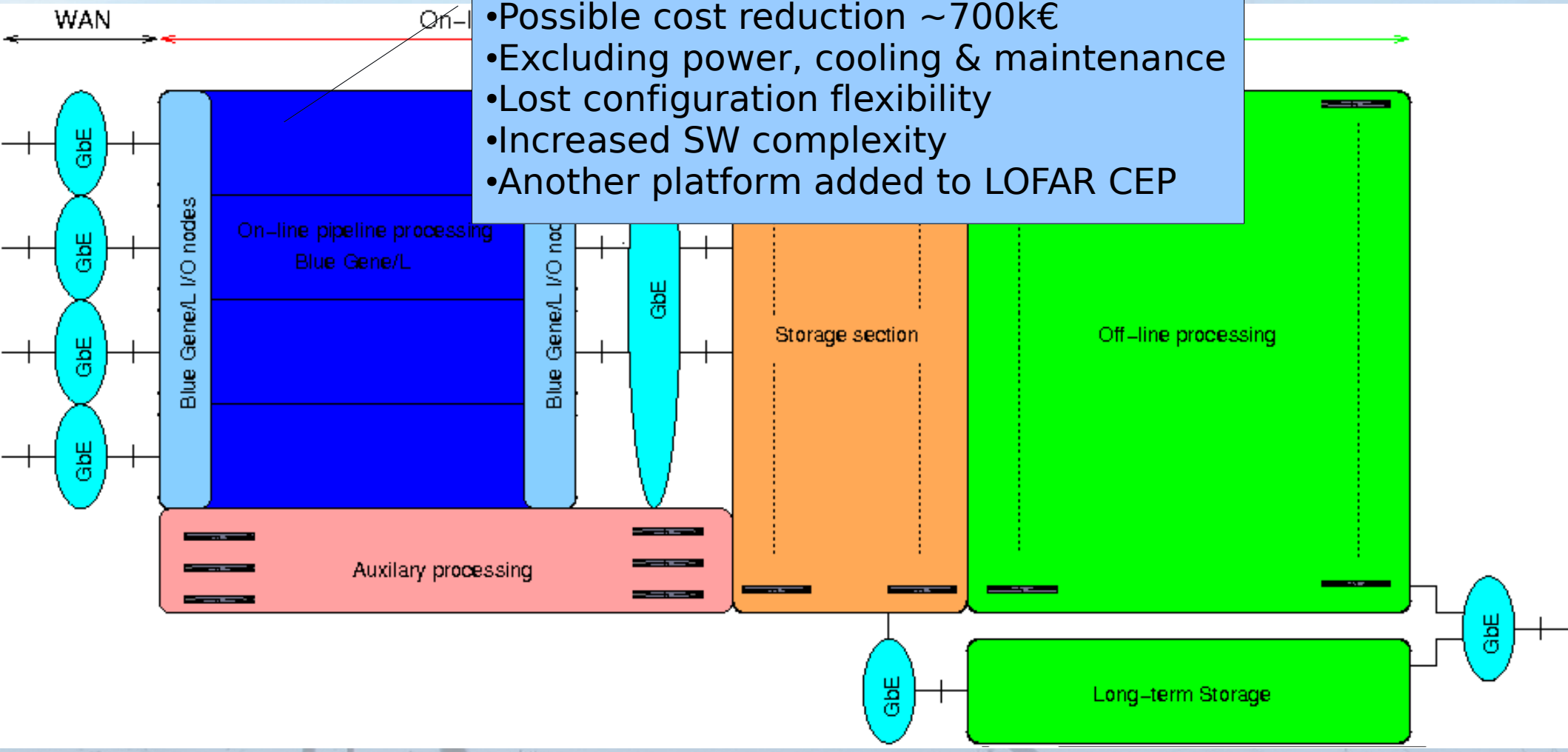
Blue Gene/L I/O

ZOID optimisations & LOFAR tasks on ION

- mem speed increase by moving area outside OS
  - avoid penalty hit by SW handling TLB misses
- second core accessible via special OS hooks
  - user responsible for ensuring disjointness
- ION now capable of replacing inputsection
  - including recv, sync and delay comp.
  - excluding network transpose

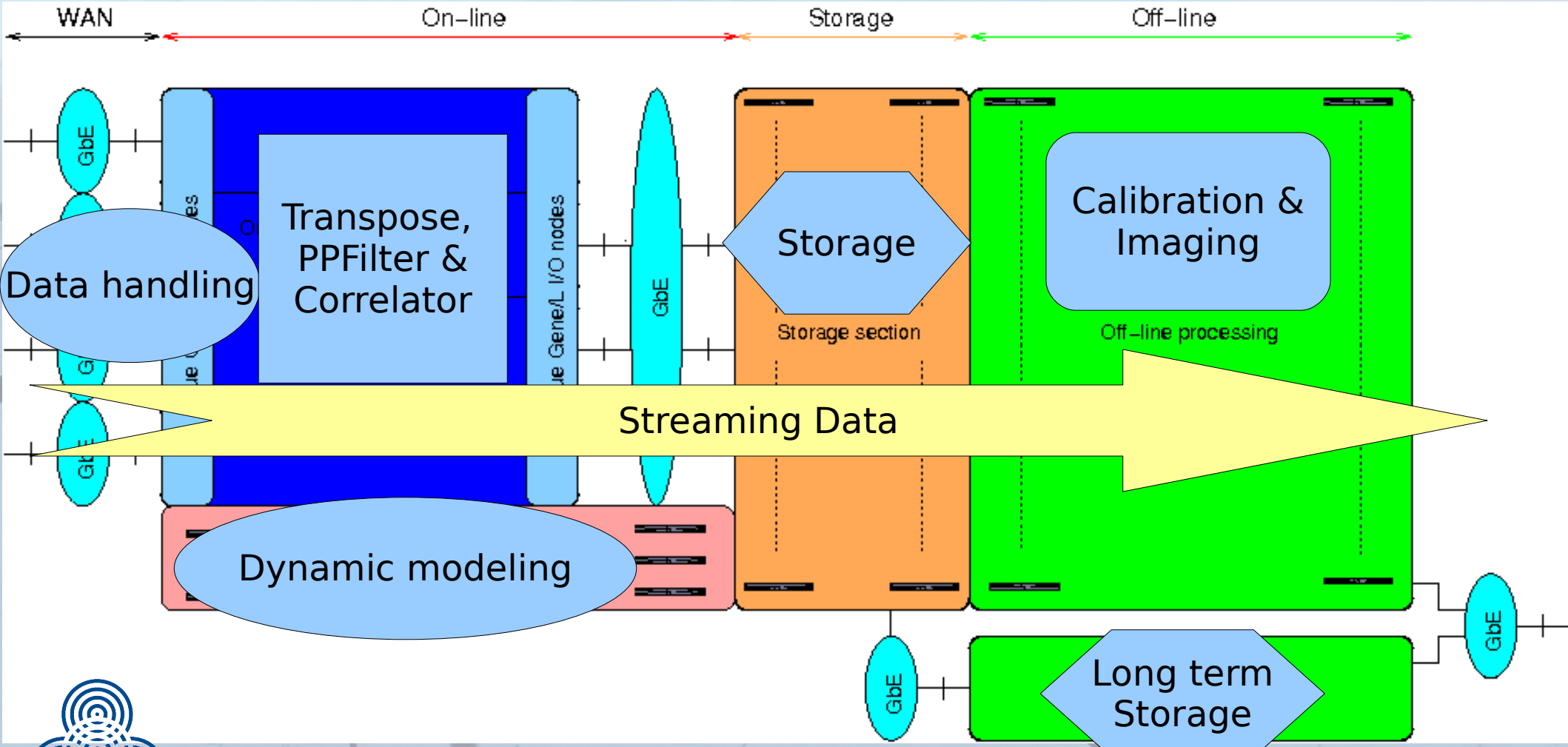
Blue Gene/L I/O

- Possible cost reduction ~700k€
- Excluding power, cooling & maintenance
- Lost configuration flexibility
- Increased SW complexity
- Another platform added to LOFAR CEP



CEP implementation model

Multiple Linux based sub-clusters, Blue Gene/L inside



Conclusions

- ZeptoOS & Zoid effectively added ~4 Tflop to our Blue Gene/L
- For streaming applications, processing on the IO node is natural
- For LOFAR the advantages seem to beat the disadvantages
- Blue Gene/P allows users on the IO node from launch

Acknowledgments

Kamil Iskra (ANL)

John W. Romein (ASTRON)

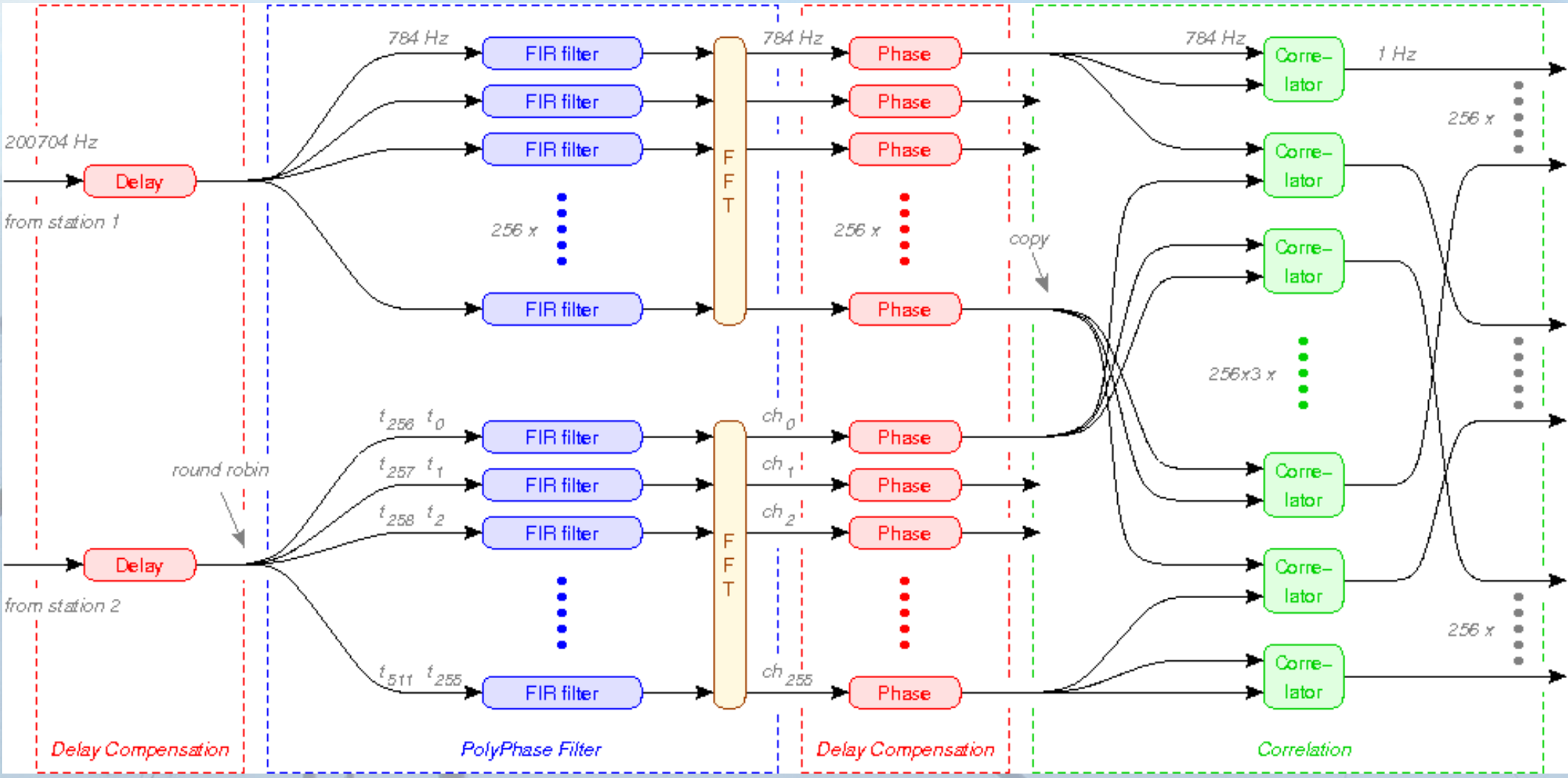
Peter Boonstoppel (VU Amsterdam)

for more information on ZeptoOS:

http://www-unix.mcs.anl.gov/zeptoos/

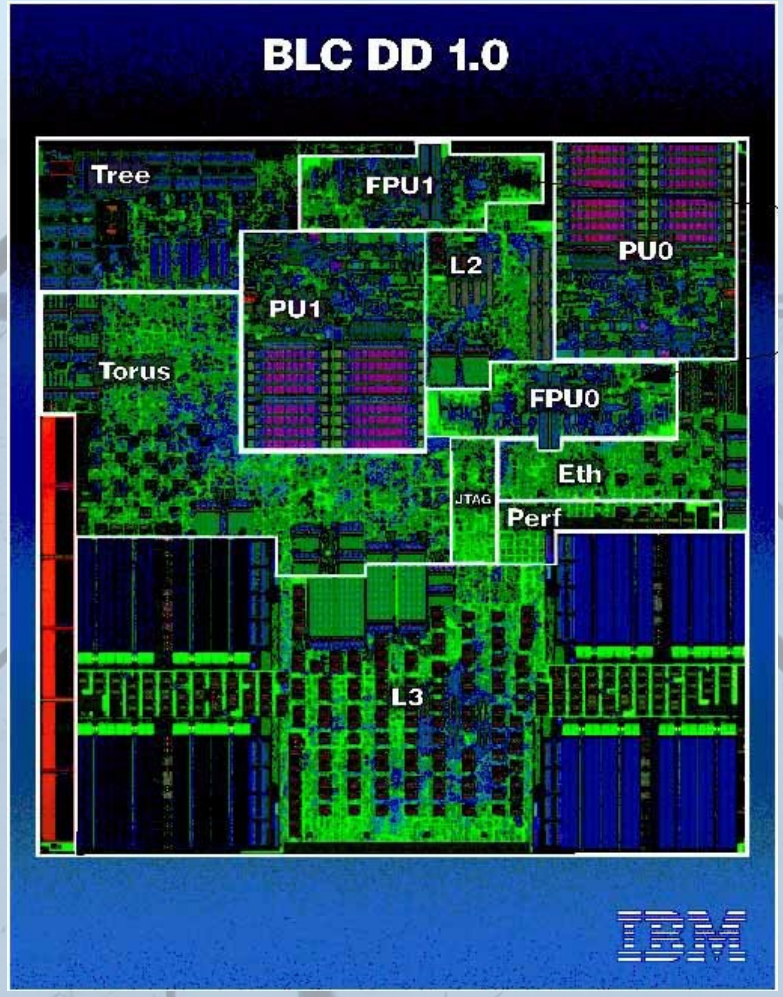
(now includes support for ZOID)

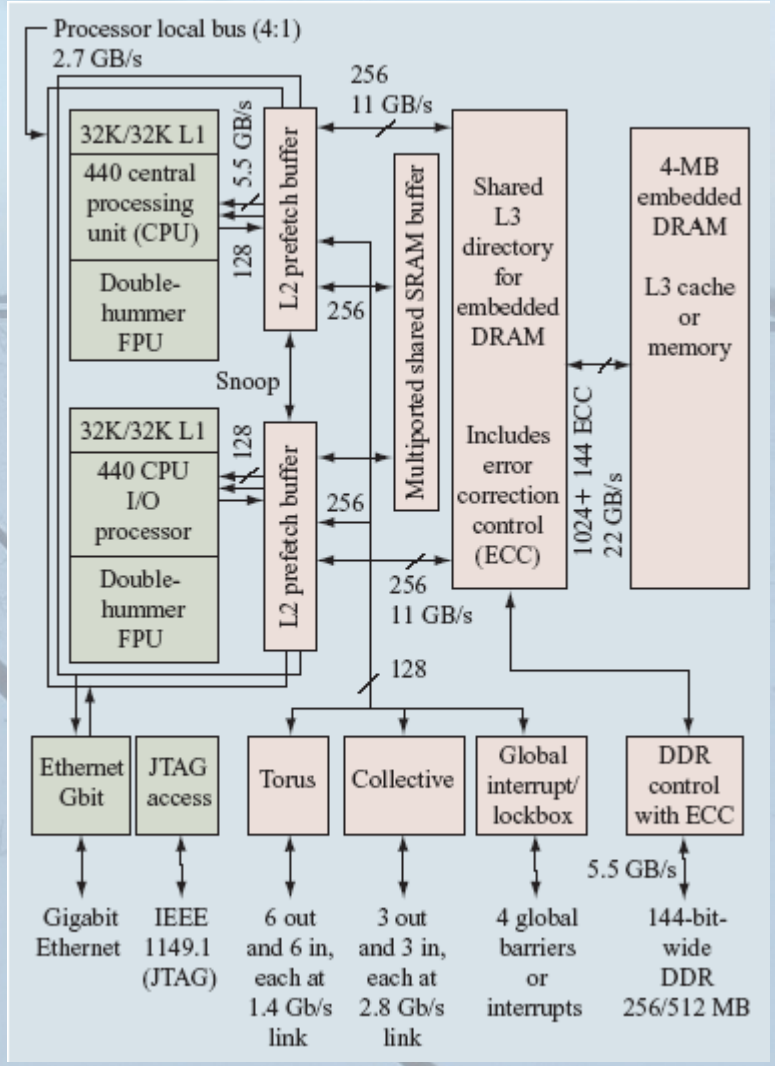
LOFAR CEP Processing



the Blue Gene/L CPU

custom, dual core, power 440 based, embedded ASIC





Blue Gene/L

Lessons learned with LOFAR

- We love the compute node ASIC
  - complex number support in assembler is great
  - the double hummer FPU performs exceptionally
  - enough registers to hide load latency (low anyway)
- But Blue Gene/L is not perfect
  - compilers are not great (but have improved greatly)
  - computational hotspots **require** assembler
  - streaming I/O is *not yet* optimal

The Next Generation
Blue Gene/P

- 850 MHz, Quad Core CPU
- Coherent L1 caches, real SMP
- factor 2.42 performance increase overall
- 10 GbE fibre vs. 1 GbE copper
- 1 ION per 16 CN vs. 1 per 8
  - 1,3 Flop/byte vs. 2,8 Flop/byte
  - DMA engine helps
- 2 GB main memory per CN

The Next Generation

Blue Gene/P

- Logical next step from Blue Gene/L
- HW differences are small
  - BGL code will run on BGP without much porting
  - Optimisations are still valid
- SW differences are all positive
  - Vendor supported access to I/O node
  - Threads / OpenMP / MPI-2 support on CN
- Performance of I/O node unproven
  - BGL ION underpowered
  - BGP ION less powerful per bps

Future trends

- Increased parallelism
- Decreased complexity
  - of core design
  - of cache structure
- Communication
  - between cores
  - into machine
- Fibre closer to core

- SKA, even bigger than LOFAR
- More processing in the field than LOFAR
- Massive correlator
- More COTS HW
- Remote locations

Blue Gene/L I/O